

IEEE CEC 2009

Ms. Pac-Man Competition

By Chan Kwong Choi Max - 13.05.2009

Overview

The software controller developed to play Ms. Pac-Man game is based on the version at <http://www.webpacman.com/mspacman.htm>

Environment

- i. Window XP Sp3
- ii. JDK 5.0
- iii. NetBeans IDE 6.5

To run the program

- i. Open <http://www.webpacman.com/mspacman.htm>
- ii. Double click "f112.jar".

Figure 1 shows two windows. The left one searches the Location of Ms. Pac-Man game. After the Game Location is found, window would be changed to the right one.

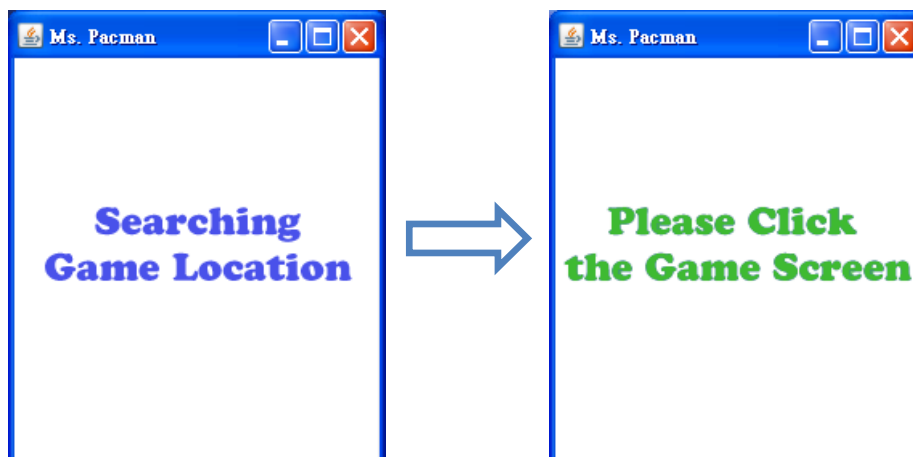


Figure 1: Two windows

- iii. Program finds the location of Ms. Pac-Man Game and starts the game automatically.

If the program cannot run properly, use NetBeans IDE to open project f112. Change Constants in Constants.java if necessary.

```
// whether to Display on JFrame Screen  
public static final boolean Display = true;
```

```
// color of Ghosts
```

```
public static final int Blinky_color = -65536;
public static final int Pinky_color = -18210;
public static final int Inky_color = -16711714;
public static final int Sue_color = -18361;

// Weak Status Color Ghost
public static final int get_blue = -14605858;

// color of Ghost's body when changing from Weak Status to Strong Status
public static final int get_gray = -2171170;

// color of Weak Ghosts' eyes
public static final int get_wg = -18281;

// to control the number of frames captured per second
public static final int Delay = 10;
```

Method used

This program consists of two parts. The first part constructs a 2D Array Road Map and finds corresponding location of Ms. Pac-Man and Ghosts. The second part makes use of those information to determine Ms. Pac-Man's next moving direction. Figure 2 shows a diagram of the program flow.

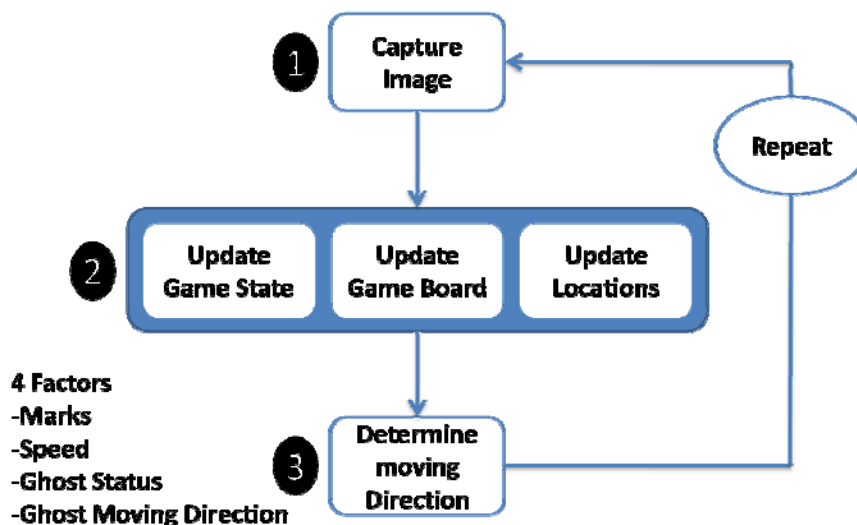


Figure 2: Diagram of Program Flow

The main idea of algorithm is to choose one direction which is safe and gives the most marks. Ghosts' Moving Direction is estimated. Figure 3 shows an example of this estimation.

$$\text{Length}_x = \text{Ghost}_x - \text{Agent}_x$$

$$\text{Length}_y = \text{Ghost}_y - \text{Agent}_y$$

$$(\text{Length}_x, \text{Length}_y)$$

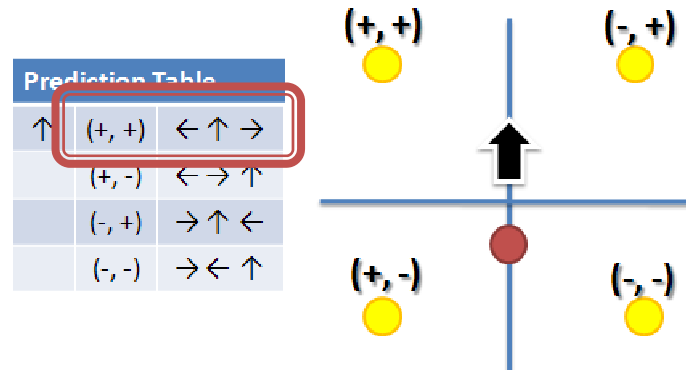


Figure 3: Example of Ghosts moving estimation

As shown in figure 3, suppose a ghost is moving upward. If Ms. Pac-Man at that moment is on the left top hand side (i.e. both Length_x & Length_y are positive), the program will assume the ghost moves left first. If it cannot move left, it will move upward. If it cannot move upward, it will move right.

With this estimation of Ghosts' moving direction, the program tries all possible moving direction of Ms. Pac-Man and sees which one is safe and gain the highest mark.